

### 3. Potisni automati

#### Zadatak 1.

Izložiti tri sekvence iz skupa sekvenci koje prepoznaje potisni automat sa jednim stanjem prikazan na Sl. **Error! No text of specified style in document..1**. Startni stek je  $\nabla$ . Prikazati rad automata za svaku od ovih sekvenci.

$S_0$ :

	a	b	c	$\dashv$
A	PUSH(A) ADVANCE	REJECT	POP ADVANCE	REJECT
B	PUSH(C) ADVANCE	POP RETAIN	PUSH(A) ADVANCE	REJECT
C	PUSH(B) ADVANCE	PUSH(C) ADVANCE	POP ADVANCE	REJECT
$\nabla$	PUSH(A) ADVANCE	PUSH(B) ADVANCE	REJECT	ACCEPT

**Sl. Error! No text of specified style in document..1**

#### Analiza problema

Potisni automat definisan je uređenom šestorkom  $(U, Y, Y_s, S, S_t, \delta_c)$  gde:

- $U$  predstavlja skup ulaznih simbola. U konkretnom slučaju, ulazni simboli označavaju pojedine kolone tabele (osim poslednje) na Sl. **Error! No text of specified style in document..1**, pa je  $U = \{a, b, c\}$ .
- $Y$  predstavlja skup simbola steka. U konkretnom slučaju, stek simboli obeležavaju pojedine vrste tabele na Sl. **Error! No text of specified style in document..1**, pa je  $Y = \{A, B, C, \nabla\}$ .
- $Y_s$  je početna konfiguracija steka i definisana je sekvencom simbola steka koja počinje simbolom  $\nabla$ . Simbol  $\nabla$  naziva se dnom steka. On se nikada ne stavlja na stek niti se sa njega skida.
- $S$  predstavlja skup stanja potisnog automata. U konkretnom slučaju,  $S = \{S_0\}$ .
- $S_t \in S$  predstavlja startno stanje automata, u konkretnom slučaju to je jedino stanje automata  $S_0$ .
- $\delta_c: (U \cup \{\dashv\}) \times Y \times S \rightarrow O$  predstavlja kontrolnu funkciju koja za svaki ulazni simbol (kojima se, iz praktičnih razloga, dodaje marker kraja ulazne sekvence  $\dashv$ ),

svaki simbol steka i svako stanje definiše određeni element skupa svih mogućih operacija potisnog automata  $O = O_y \times S \times O_u \cup \{\text{ACCEPT}, \text{REJECT}\}$ , pri čemu:

- $O_y = \{\text{POP}, \text{PUSH}(x), \text{nepromenjen stek}\}$  je skup mogućih operacija nad stekom, POP označava skidanje jednog simbola sa steka, PUSH(x) označava stavljanje simbola steka x na vrh steka.
- $O_u = \{\text{ADVANCE}, \text{RETAIN}\}$  je skup mogućih operacija nad ulaznom sekvencom. ADVANCE označava da sledeći ulazni simbol iza tekućeg postaje novi tekući simbol, a RETAIN označava da se tekući simbol ulaza ne menja.

Prema tome, svaka operacija automata sastoji se iz određene operacije nad stekom, promene stanja u određeno stanje  $s \in S$  i određene operacije nad ulazom. Izuzetak od ovoga su operacije ACCEPT koja označava kraj rada automata pri čemu se ulazna sekvenca prihvata i REJECT koja označava kraj rada pri čemu se ulazna sekvenca odbija. Kontrolna funkcija  $\delta_c$  je u konkretnom slučaju zadata kontrolnom tabelom na Sl. **Error! No text of specified style in document..1**.

Rad automata odvija se po sledećem algoritmu:

```

tekuće_stanje :=  $S_t$ ;
tekući_ulaz := prvi simbol ulazne sekvence;
tekuća_konfiguracija_steka :=  $Y_s$ ;
loop
    if  $\delta_c(\text{tekući\_ulaz}, \text{tekuća\_konfiguracija\_steka}, \text{tekuće\_stanje}) = \text{ACCEPT}$ 
        then kraj rada uz prihvatanje ulazne sekvence;
    else if  $\delta_c(\text{tekući\_ulaz}, \text{tekuća\_konfiguracija\_steka}, \text{tekuće\_stanje}) =$ 
REJECT
        then kraj rada uz odbijanje ulazne sekvence
    else
         $\delta_c(\text{tekući\_ulaz}, \text{tekuća\_konfiguracija\_steka}, \text{tekuće\_stanje})$  je oblika (y,
s, u);
        primeniti na tekuću_konfiguraciju_steka operaciju y;
        tekuće_stanje := s;
        primeniti na tekući_ulaz operaciju u;
    end if
end loop;

```

Skup svih sekvenci ulaznih simbola koje automat prihvata naziva se jezik tog automata.

#### Rešenje

a)

Primeri sekvenci koje prihvata gore dati potisni automat:

1) prazna ulazna sekvenca:  $\dashv$

stek:

ulaz:

akcija:

1.	$\nabla$	$\neg$	ACCEPT
----	----------	--------	--------

2) ulazna sekvenca:  $ac\neg$

stek:	ulaz:	akcija:
1. $\nabla$	$ac\neg$	PUSH(A), ADVANCE
2. $\nabla A$	$c\neg$	POP, ADVANCE
3. $\nabla$	$\neg$	ACCEPT

3) ulazna sekvenca:  $aaccac\neg$

stek:	ulaz:	akcija:
1. $\nabla$	$aaccac\neg$	PUSH(A), ADVANCE
2. $\nabla A$	$accac\neg$	PUSH(A), ADVANCE
3. $\nabla AA$	$ccac\neg$	POP, ADVANCE
4. $\nabla A$	$cac\neg$	POP, ADVANCE
5. $\nabla$	$ac\neg$	PUSH(A), ADVANCE
6. $\nabla A$	$c\neg$	POP, ADVANCE
7. $\nabla$	$\neg$	ACCEPT

### Napomena:

Navedeni skup osnovnih operacija za rad sa stekom se može proširiti operacijom REPLACE. Operacija REPLACE(XYZ) ima efekat kao sledeći niz osnovnih operacija: POP, PUSH(X), PUSH(Y), PUSH(Z). dakle, prvo skida vršni simbol sa steka, a zatim stavlja na stek simbole zadate u argumentu redom kojim su navedeni.

### Zadatak 2. (3.1.4 – deo)

Projektovati potisni automat koji prepoznaje sledeći skup sekvenci:

$$\{1^n 0^m\} \quad n > m > 0$$

Iz zadatog skupa izabrati jednu sekvencu dužine veće od tri i prikazati proces njenog prepoznavanja.

### Rešenje

Ideja konstrukcije automata je sledeća: mada se svaki potisni automat može napraviti sa jednim stanjem, primera radi ćemo ovaj realizovati sa dva stanja. U početnom stanju  $S_1$  automat će obrađivati vodeće jedinice na ulazu tako što će za svaku jedinicu na stek, koji je inicijalno prazan, biti stavljen simbol A. U trenutku kada se na ulazu pojavi nula,

automat prelazi u stanje  $S_2$  koje označava da se u nastavku prihvataju isključivo nule. Pri obradi nule, sa steka se skida jedan simbol A. Sekvenca se prihvata jedino ako se stek ne isprazni do pojave markera kraja ulaza, jer to označava da je broj jedinica na ulazu bio veći od broja nula. Konstrukciju sprovedimo po koracima:

1. Uočavamo ulazne simbole:  $\{0, 1\}$
2. Uočavamo simbole steka:  $\{\nabla, A\}$
3. Uočavamo simbole stanja:  $\{S_1, S_2\}$
4. Definišemo početnu konfiguraciju steka  $\nabla$ .
5. Pravimo kontrolne tabele (Sl. Error! No text of specified style in document..2).

Startni stek: $\nabla$		$S_1$		$S_2$	
		0	1	$\neg$	
A	POP	ADVANCE STATE ( $S_2$ )	PUSH(A) ADVANCE STATE( $S_1$ )	REJECT	ACCEPT
	ADVANCE STATE ( $S_2$ )				
$\nabla$	REJECT	PUSH(A) ADVANCE STATE( $S_1$ )	REJECT	REJECT	REJECT
	ADVANCE STATE ( $S_1$ )				

Sl. Error! No text of specified style in document..2

Prikazaćemo proces prepoznavanja sekvence 11100:

stek:	ulaz:	akcija:	stanje:
1. $\nabla$	11100 $\neg$	PUSH(A), ADVANCE, STATE( $S_1$ )	$S_1$
2. $\nabla A$	1100 $\neg$	PUSH(A), ADVANCE, STATE( $S_1$ )	$S_1$
3. $\nabla AA$	100 $\neg$	PUSH(A), ADVANCE, STATE( $S_1$ )	$S_1$
4. $\nabla AAA$	00 $\neg$	POP, ADVANCE, STATE( $S_2$ )	$S_1$
5. $\nabla AA$	0 $\neg$	POP, ADVANCE, STATE( $S_2$ )	$S_2$
6. $\nabla A$	$\neg$	ACCEPT	$S_2$

## Parsiranje od dna ka vrhu

### Zadatak 1.

Parser od dna ka vrhu napravljen na osnovu sledeće gramatike prepoznaje smene u zadatoj ulaznoj sekvenci po sledećem redosledu:

4, 4, 4, 1, 4, 4, 1, 3, 2

- a) O kojoj ulaznoj sekvenci je reč?  
b) Prikazati rad parsera za ulaznu sekvencu iz tačke a).

1.  $\langle S \rangle \rightarrow \langle S \rangle \langle S \rangle a$
2.  $\langle S \rangle \rightarrow \langle S \rangle b$
3.  $\langle S \rangle \rightarrow \langle S \rangle \langle S \rangle \langle S \rangle c$
4.  $\langle S \rangle \rightarrow d$

### Rešenje

a)

- redosled smena je obrnut krajnje desnom izvođenju

$$\begin{aligned} \langle S \rangle &\Rightarrow_{rm} \langle S \rangle b \Rightarrow_{rm} \langle S \rangle \langle S \rangle \langle S \rangle cb \Rightarrow_{rm} \langle S \rangle \langle S \rangle \langle S \rangle \langle S \rangle acb \Rightarrow_{rm} \langle S \rangle \langle S \rangle \langle S \rangle \langle S \rangle dacb \Rightarrow_{rm} \\ &\quad \uparrow \quad \quad \uparrow \quad \quad \quad \uparrow \quad \quad \quad \uparrow \quad \quad \quad \uparrow \\ &\quad 2 \quad \quad 3 \quad \quad \quad 1 \quad \quad \quad 4 \quad \quad \quad 4 \\ &\Rightarrow_{rm} \langle S \rangle \langle S \rangle ddacb \Rightarrow_{rm} \langle S \rangle \langle S \rangle \langle S \rangle addacb \Rightarrow_{rm} \langle S \rangle \langle S \rangle \langle S \rangle daddacb \Rightarrow_{rm} \\ &\quad \quad \quad \uparrow \quad \quad \quad \uparrow \quad \quad \quad \uparrow \\ &\quad \quad 1 \quad \quad \quad 4 \quad \quad \quad 4 \\ &\Rightarrow_{rm} \langle S \rangle ddaddacb \Rightarrow_{rm} dddaddacb \\ &\quad \quad \quad \uparrow \\ &\quad \quad 4 \end{aligned}$$

b)

- stek + ulaz = sentencijalna forma u krajnje desnom izvođenju ulaza iz  $\langle S \rangle$
- parser potiskuje simbole sa ulaza na stek dok ne kompletira ručku (desnu stranu smene koja je poslednja primenjena u desnom izvođenju posmatrane sentencijalne forme)
- tada se ručka zamenjuje levom stranom smene.

Stanje steka :	Ulazni simbol :	Akcija
1. $\nabla$	dddaddacb	SHIFT
2. $\nabla \underline{d}$	ddaddacb	REDUCE(4)
3. $\nabla \langle S \rangle$	ddaddacb	SHIFT
4. $\nabla \langle S \rangle \underline{d}$	daddacb	REDUCE(4)
5. $\nabla \langle S \rangle \langle S \rangle$	daddacb	SHIFT

6. $\nabla \langle S \rangle \langle S \rangle \underline{d}$	addacb	REDUCE(4)
7. $\nabla \langle S \rangle \langle S \rangle \langle S \rangle$	addacb	SHIFT
8. $\nabla \langle S \rangle \underline{\langle S \rangle \langle S \rangle} a$	ddacb	REDUCE(1)
9. $\nabla \langle S \rangle \langle S \rangle$	ddacb	SHIFT
10. $\nabla \langle S \rangle \langle S \rangle \underline{d}$	dacb	REDUCE(4)
11. $\nabla \langle S \rangle \langle S \rangle \langle S \rangle$	dacb	SHIFT
12. $\nabla \langle S \rangle \langle S \rangle \langle S \rangle \underline{d}$	acb	REDUCE(4)
13. $\nabla \langle S \rangle \langle S \rangle \langle S \rangle \langle S \rangle$	acb	SHIFT
14. $\nabla \langle S \rangle \langle S \rangle \underline{\langle S \rangle \langle S \rangle} a$	cb	REDUCE(1)
15. $\nabla \langle S \rangle \langle S \rangle \langle S \rangle$	cb	SHIFT
16. $\nabla \underline{\langle S \rangle \langle S \rangle \langle S \rangle} c$	b	REDUCE(3)
17. $\nabla \langle S \rangle$	b	SHIFT
18. $\nabla \underline{\langle S \rangle} b$		REDUCE(2)
19. $\nabla \langle S \rangle$		ACCEPT

Ručke su prikazane podebljano i podvučeno. U slučaju da na steku nije ručka vrši se ubacivanje novog simbola na stek.

SHIFT  $\equiv$  PUSH(tekući ulazni simbol), ADVANCE

REDUCE(1)  $\equiv$  POP, POP, POP, PUSH( $\langle S \rangle$ )

REDUCE(2)  $\equiv$  POP, POP, PUSH( $\langle S \rangle$ )

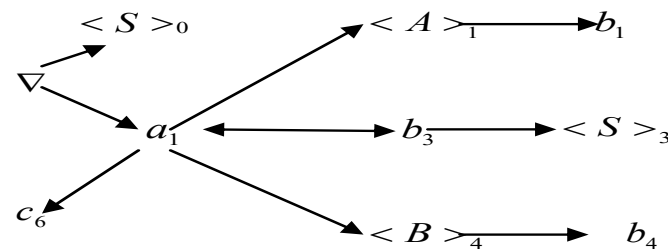
REDUCE(3)  $\equiv$  POP, POP, POP, POP, PUSH( $\langle S \rangle$ )

REDUCE(4)  $\equiv$  POP, PUSH( $\langle S \rangle$ )

### Zadatak 2.

Data je gramatika sa startnim simbolom  $\langle S \rangle$  i deo grafa prelaza odgovarajućeg potisnog automata. Nacrtati izgled steka i stablo izvođenja kada "potisni-svedi" parser procesira niz znakova *abacbbb*.

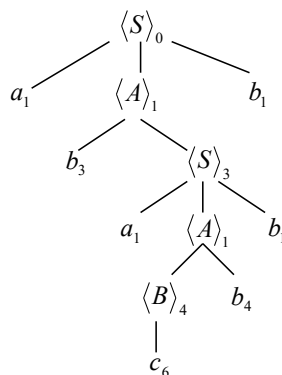
1.  $\langle S \rangle \rightarrow a \langle A \rangle b$
2.  $\langle S \rangle \rightarrow c$
3.  $\langle A \rangle \rightarrow b \langle S \rangle$
4.  $\langle A \rangle \rightarrow \langle B \rangle b$
5.  $\langle B \rangle \rightarrow a \langle A \rangle$
6.  $\langle B \rangle \rightarrow c$



Graf prelaza potisnog automata "potisni-svedi"

## Rešenje

Stek	Ulazni niz	Operacija mašine
1. $\nabla$	abacbbb	SHIFT
2. $\nabla a_1$	bacbbb	SHIFT
3. $\nabla a_1 b_3$	acbbb	SHIFT
4. $\nabla a_1 b_3 a_1$	cbbb	SHIFT
5. $\nabla a_1 b_3 a_1 c_6$	bbb	REDUCE(6)
6. $\nabla a_1 b_3 a_1 <B>_4$	bbb	SHIFT
7. $\nabla a_1 b_3 a_1 <B>_4 b_4$	bb	REDUCE(4)
8. $\nabla a_1 b_3 a_1 <A>_1$	bb	SHIFT
9. $\nabla a_1 b_3 a_1 <A>_1 b_1$	b	REDUCE(1)
10. $\nabla a_1 b_3 <S>_3$	b	REDUCE(3)
11. $\nabla a_1 <A>_1$	b	SHIFT
12. $\nabla a_1 <A>_1 b_1$	—	REDUCE(1)
13. $\nabla <S>_0$	—	ACCEPT



- odluka o sledećoj akciji parsera donosi se isključivo na osnovu vršnog simbola steka i (eventualno) tekućeg ulaznog simbola.

SHIFT: PUSHT(*tekući ulazni simbol*), ADVANCE

REDUCE( $<X> \rightarrow A_1 A_2 \dots A_n$ ):  $n$  puta akcija POP, PUSHT( $<X>$ )

- PUSHT( $X$ ) konsultuje posebnu, takozvanu potisnu tabelu, da na osnovu vršnog simbola steka i simbola  $X$  odredi koje gramatičko dešavanje simbola  $X$  treba potisnuti na stek. Potisna tabela se može alternativno predstaviti grafom prelaza koji prikazuje koji symbol treba staviti na stek u zavisnosti od trenutnog vrha steka i ulaza. U grafu prelaza čvorove predstavljaju gramatička dešavanja u gramatici, dok su prelazi određeni ulaznim simbolima.

## Diskusija

- U opštem slučaju, simboli steka odgovaraju skupovima gramatičkih dešavanjima a ne samo pojedinačnim dešavanjima.

Na primer, ako bismo gramatici dodali smenu:

7.  $<B> \rightarrow b c$

tada za datu ulaznu sekvencu ab..., kada treba potisnuti  $b$  na stek, ne zna se da li je to  $b_3$  ili  $b_7$  (iz 3. ili 7. smene). U tom slučaju postojao bi simbol steka  $b_x$  koji bi odgovarao skupu gramatičkih dešavanja  $\{b_3, b_7\}$ .

## Zadatak 3. PRIMER RAĐEN NA PREDAVANJIMA

Za datu gramatiku konstruisati

- Karakteristični LR(0) automat konfiguracionim metodom.
- LR(0) automat
- SLR(1) automat

1.  $<S> \rightarrow a$

2.  $<S> \rightarrow ( <S> <R>$

3.  $<R> \rightarrow , <S> <R>$

4.  $<R> \rightarrow )$

Rešenje:

a) Konstrukcija karakterističnog automata i odgovarajuće potisne tabele

## Analiza problema

- LR(0) konfiguracije 2. smene

$<S> \rightarrow \bullet ( <S> <R>$  konfiguracija zatvaranja

bazične konfiguracije:

$<S> \rightarrow ( \bullet <S> <R>$   $<S> \rightarrow ( <S> \bullet <R>$   $<S> \rightarrow ( <S> <R> \bullet$

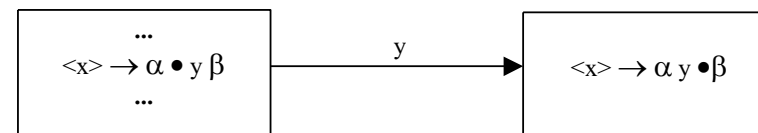
- za praznu smenu  $<X> \rightarrow \epsilon$  jedna konfiguracija, u oznaci  $<X> \rightarrow \epsilon \bullet$  ili  $<X> \rightarrow \bullet$ .
- proširenje gramatike (uniformiše pravila konstrukcije parsera)
  - $<S'> \rightarrow <S> \mid$

- Konstrukcija karakterističnog LR(0) automata (stanja su skupovi konfiguracija)

1. startno stanje  $\text{Closure}_0(\{<S'> \rightarrow \bullet <S> \mid\})$ .

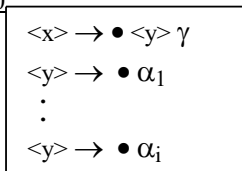
2. prelaze definiše operacija Goto0. Na novodobijena stanja uvek primeniti  $\text{Closure}_0$ .

## Goto0



## Closure0

dodaje se



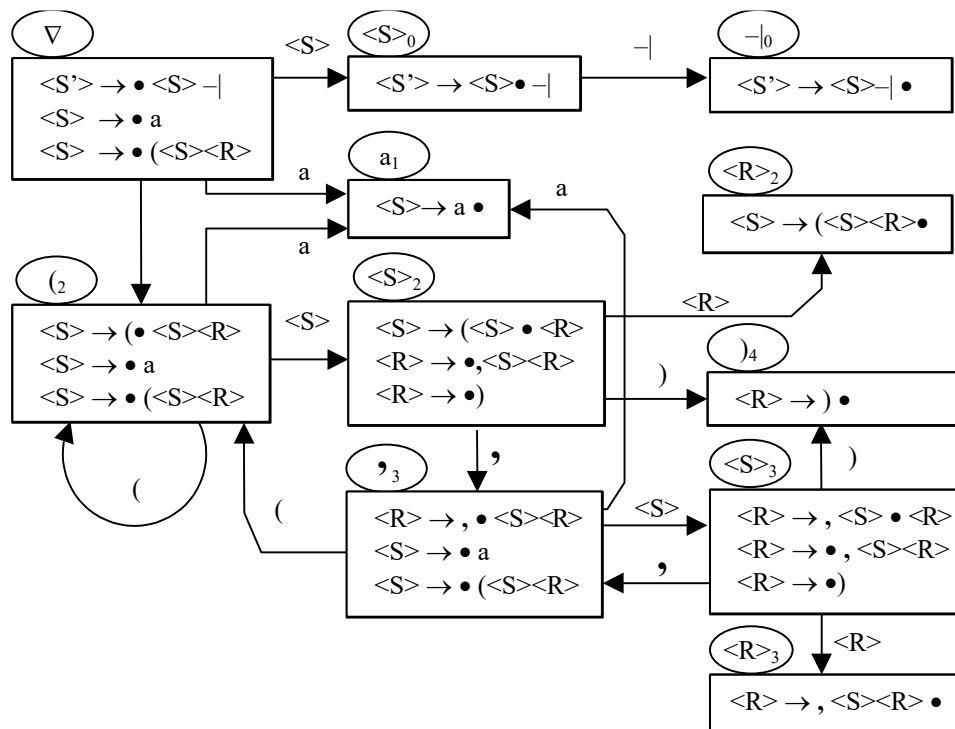
iterativno

$<x> \rightarrow \bullet <y> \dots$

$<y> \rightarrow \bullet <z> \dots$

$<z> \rightarrow \bullet \dots$

Karakteristični LR(0) automat



Potisna tabela LR(0) automata

	$\langle S \rangle$	$\langle R \rangle$	$($	$,$	$)$	$a$	$- $
$\nabla$	$\langle S \rangle_0$		$(2$			$a_1$	
$\langle S \rangle_0$							$- _0$
$- _0$							
$a_1$							
$(2$	$\langle S \rangle_2$		$(2$			$a_1$	
$\langle S \rangle_2$		$\langle R \rangle_2$		$,_3$	$)_4$		
$\langle R \rangle_2$							
$,_3$	$\langle S \rangle_3$		$(2$			$a_1$	
$\langle S \rangle_3$		$\langle R \rangle_3$		$,_3$	$)_4$		
$\langle R \rangle_3$							
$)_4$							

b) Kontrolna tabela LR(0) automata

#### Analiza problema

Kontrolna tabela LR(0) parsera se dobija na osnovu LR(0) potisne tabele

Pravila za popunjavanje LR(0) kontrolne tabele na osnovu karakterističnog automata su sledeća (razmatra se ulaz u vrsti V, kojoj odgovara skup konfiguracija S):

1. Ako S sadrži konfiguraciju oblika  $\langle X \rangle \rightarrow \alpha \bullet x \beta$ , ulaz treba popuniti akcijom SHIFT.
2. Ako S sadrži konfiguraciju oblika  $\langle Y \rangle \rightarrow \gamma \bullet$ , odnosno tačka se nalazi na kraju desne strane, ulaz treba popuniti akcijom REDUCE( $\langle Y \rangle \rightarrow \gamma$ ). Izuzetno, ako se radi o nultoj smeni, akcija je ACCEPT.
3. Vrste koje nisu pokrivene ni 1. ni 2. pravilom treba da sadrže akciju REJECT.

Ukoliko gornja pravila na jednoznačan način određuju akciju za svaki ulaz kontrolne tabele, gramatika pripada klasi LR(0).

	Akcija
$\nabla$	SHIFT
$\langle S \rangle_0$	SHIFT
$- _0$	ACCEPT
$a_1$	REDUCE(1)
$(_2$	SHIFT
$\langle S \rangle_3$	SHIFT
$\langle R \rangle_2$	REDUCE(2)
$,_3$	SHIFT
$\langle S \rangle_6$	SHIFT
$\langle R \rangle_3$	REDUCE(3)
$)_4$	REDUCE(4)

c) Kontrolna tabela SLR(1) automata

#### Analiza problema

- Kontrolna tabela SLR(1) parsera se dobija na osnovu LR(0) potisne tabele
- Popunjavanje kontrolne tabele (razmatra se ulaz u vrsti V, kojoj odgovara skup konfiguracija S i koloni x):

1. Ako S sadrži konfiguraciju oblika  $\langle X \rangle \rightarrow \alpha \bullet x \beta$ , odnosno tačka se nalazi ispred terminalnog simbola x, ulaz treba popuniti akcijom SHIFT.
2. Ako S sadrži konfiguraciju oblika  $\langle Y \rangle \rightarrow \gamma \bullet$ , odnosno tačka se nalazi na kraju desne strane, a važi da je  $x \in \text{FOLLOW}(\langle Y \rangle)$ , tada ulaz treba popuniti akcijom REDUCE( $\langle Y \rangle \rightarrow \gamma$ ). Izuzetno, ako se radi o nultoj smeni, akcija je ACCEPT.
3. Vrste koje nisu pokrivene ni 1. ni 2. pravilom treba da sadrže akciju REJECT.

Ukoliko gornja pravila na jednoznačan način određuju akciju za svaki ulaz kontrolne tabele, gramatika pripada klasi SLR(1).

#### Rešenje

$\text{FOLLOW}(\langle S' \rangle) = \{ \text{---} \}$   $\text{FOLLOW}(\langle S \rangle) = \{ , , \text{---} \}$   $\text{FOLLOW}(\langle R \rangle) = \{ , , \text{---} \}$

	(	,	)	a	---
$\nabla$	SHIFT			SHIFT	
$\langle S \rangle_0$					SHIFT
$\text{---}_0$					ACCEPT
$a_1$		REDUCE(1)	REDUCE(1)		REDUCE(1)
$(_2$	SHIFT			SHIFT	
$\langle S \rangle_2$		SHIFT	SHIFT		
$\langle R \rangle_2$		REDUCE(2)	REDUCE(2)		REDUCE(2)
$,_3$	SHIFT			SHIFT	
$\langle S \rangle_3$		SHIFT	SHIFT		
$\langle R \rangle_3$		REDUCE(3)	REDUCE(3)		REDUCE(3)
$)_4$		REDUCE(4)	REDUCE(4)		REDUCE(4)

#### Zadatak 4. RAĐEN NA PREDAVANJIMA

Za datu gramatiku konstruisati LALR(1) potisni automat.

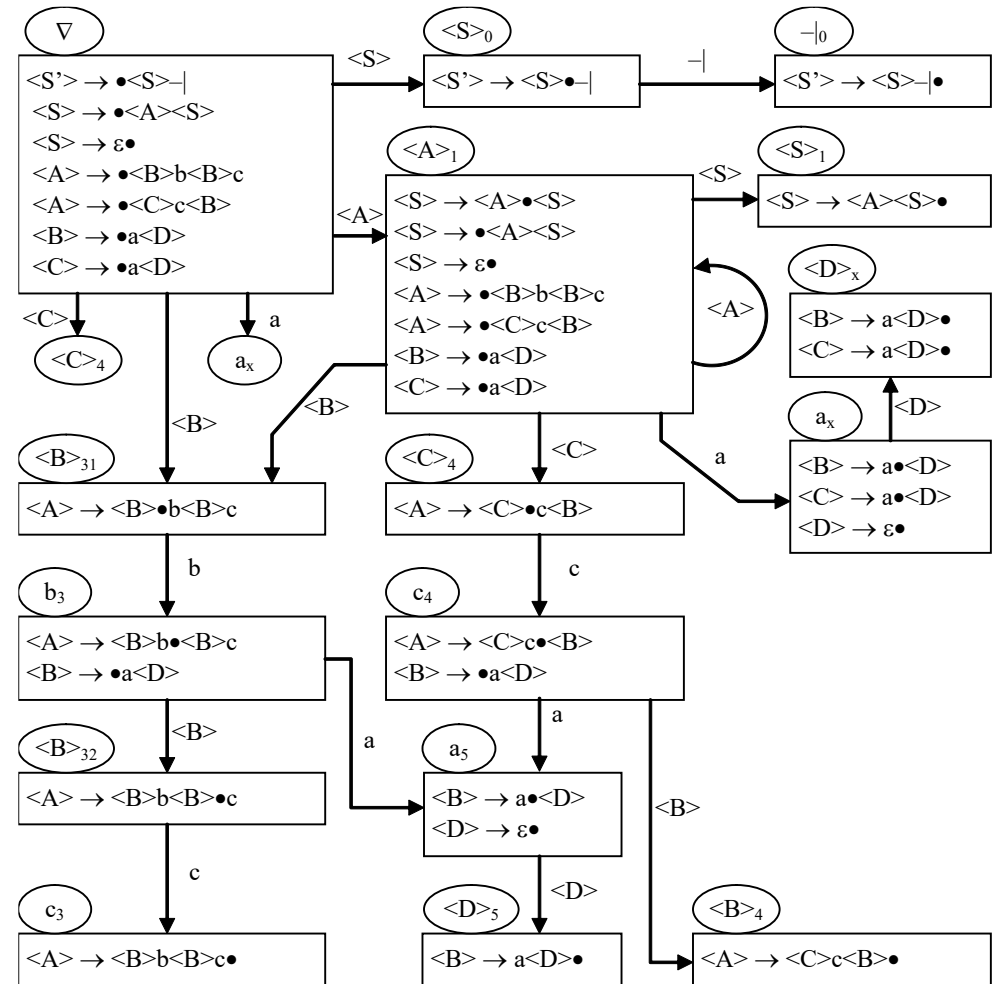
- $\langle S \rangle \rightarrow \langle A \rangle \langle S \rangle$
- $\langle S \rangle \rightarrow \varepsilon$
- $\langle A \rangle \rightarrow \langle B \rangle b \langle B \rangle c$
- $\langle A \rangle \rightarrow \langle C \rangle c \langle B \rangle$
- $\langle B \rangle \rightarrow a \langle D \rangle$
- $\langle C \rangle \rightarrow a \langle D \rangle$
- $\langle D \rangle \rightarrow \varepsilon$

#### Analiza problema

Konstrukcija se obavlja u tri koraka: 1. određivanje karakterističnog LR(0) automata, 2. određivanje predikcionih skupova i 3. konstrukcija potisne i kontrolne tabele parsersa.

#### Određivanje karakterističnog LR(0) automata

Stanjima karakterističnog LR(0) automata (detektora ručki) odgovaraju skupovi LR(0) konfiguracija, a ulazi automata određeni su simbolima gramatike proširene 0. smenom. Izgled karakterističnog LR(0) automata za datu gramatiku je sledeći:



#### Određivanje predikcionih skupova

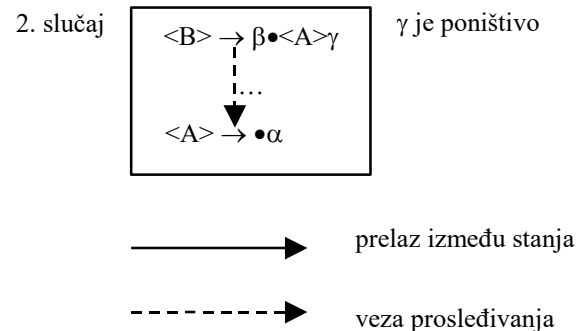
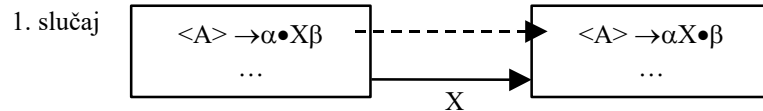
Za dobijanje LALR(1) automata potrebno je odrediti predikcione skupove LR(0) konfiguracija u pojedinim stanjima LR(0) automata. U cilju računanja predikcionih skupova prvo se odrede takozvani skupovi spontaninih predikcionih simbola (skr. SLA) za sve konfiguracije zatvaranja u svakom od stanja LR(0) detektora ručki. SLA skup za proizvoljnu LR(0) konfiguraciju zatvaranja  $\langle A \rangle \rightarrow \bullet \alpha$  u stanju  $s$  računa se tako što se nađu sve konfiguracije oblika  $\langle B \rangle \rightarrow \beta \bullet \langle A \rangle \gamma$  u stanju  $s$  i u skup  $\text{SLA}(\langle A \rangle \rightarrow \bullet \alpha)$  uključe svi terminali  $t \in \text{FIRST}(\gamma)$ .

Između LR(0) konfiguracija u stanjima LR(0) automata definišu se *veze prosleđivanja* prema sledećim pravilima:

- Za svaku konfiguraciju oblika  $\langle A \rangle \rightarrow \alpha \bullet X \beta$  u posmatranom stanju  $S$ , gde su  $\alpha$  i  $\beta$  proizvoljne sekvence od nula ili više gramatičkih simbola, a  $X$  proizvoljan gramatički

simbol, uvodi se veza prosleđivanja ka konfiguraciji oblika  $\langle A \rangle \rightarrow \alpha X \bullet \beta$  u stanju u koje se prelazi iz S po X.

2. Za svaku konfiguraciju oblika  $\langle B \rangle \rightarrow \beta \bullet \langle A \rangle \gamma$  u posmatranom stanju S, gde je  $\beta$  proizvoljna sekvenca od nula ili više gramatičkih simbola,  $\gamma$  je poništiva sekvenca od nula ili više gramatičkih simbola, a  $\langle B \rangle$  je proizvoljan neterminal, uvode se veze prosleđivanja ka svim konfiguracijama oblika  $\langle A \rangle \rightarrow \bullet \alpha$  u posmatranom stanju S.



**SL3:** Pravila prosleđivanja predikcionih simbola

Predikcioni (skr. LA) skupovi računaju se sledećim algoritmom:

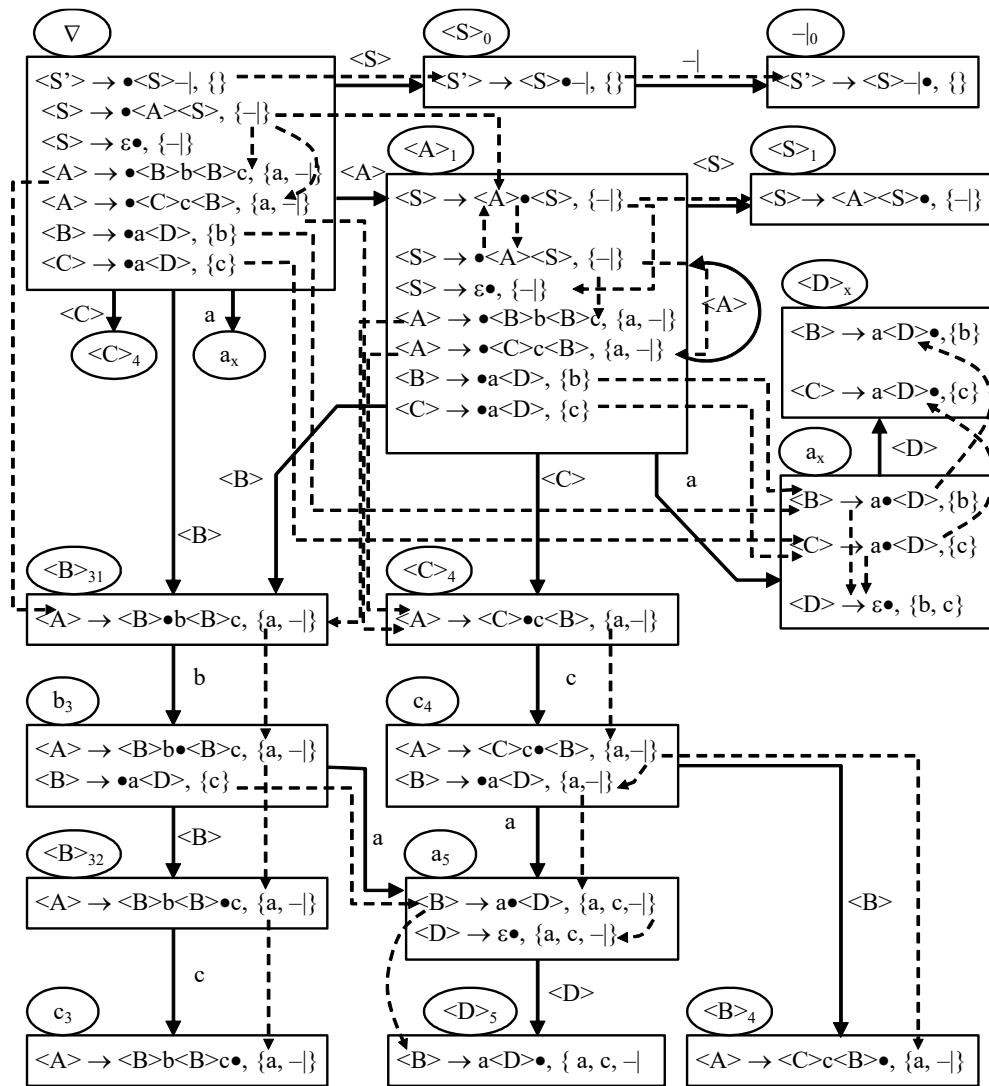
Inicijalno su LA skupovi konfiguracija zatvaranja u svakom od stanja jednaki SLA skupovima odgovarajućih konfiguracija, dok su LA skupovi ostalih konfiguracija prazni. Algoritam koristi stek na koji se stavljaju trojke (stanje, konfiguracija, predikcioni simbol). Inicijalno na stek idu sve trojke koje odgovaraju konfiguracijama zatvaranja iz svih stanja i simbolima iz njihovih SLA skupova. Trojke sa steka se redom razmatraju, pri čemu se ažuriraju LA skupovi konfiguracija koje su povezane vezama prosleđivanja sa razmatranom konfiguracijom. Na stek idu novo dodati elementi ažuriranih LA skupova. Postupak se završava kada se stek isprazni.

```

ulaz: LR(0) automat sa vezama prosleđivanja između konfiguracija,
      SLA skupovi svih konfiguracija zatvaranja u svim stanjima;
izlaz: LA skupovi svih konfiguracija u svim stanjima.
stek je inicijalno prazan;
for (∀ stanje S)
  for (∀ konfiguraciju K ∈ S)
    if (K je oblika <A> → • α)
      then LA(K) := SLA(K);
      for (t ∈ SLA(K))
        push( S, K, t);
      end for;
    else LA(K) := ∅;
  end if
end for;
while (stek nije prazan)
  pop( S, K, t);
  for (∀ konfiguraciju K1 ka kojoj ide veza prosleđivanja od K)
    if ( t ∉ LA(K1))
      then dodati t u LA(K1);
      S1 := stanje automata u kome se nalazi K1;
      push(S1, K1, t);
    end if;
  end for;
end while;

```

Po završetku rada algoritma, dobija se LALR(1) detektor ručki prikazan na Slici 2.



Sl. Error! No text of specified style in document.4: Ilustracija iterativnog određivanja predikcionih simbola

### Konstrukcija parserskih tabela

Tabelarno prikazan LALR(1) automat predstavlja potisnu tabelu parsera. Vrste kontrolne tabele LALR(1) parsera odgovaraju pojedinim stanjima karakterističnog automata, a kolone pojedinim ulaznim simbolima gramatike proširene 0. smenom. Pravila popunjavanja LALR(1) kontrolne tabele na osnovu karakterističnog LALR(1) automata su sledeća (razmatra se ulaz u vrsti V, kojoj odgovara skup konfiguracija S i koloni x):

1. Ako S sadrži konfiguraciju oblika  $\langle X \rangle \rightarrow \alpha \bullet x \beta$ , t odnosno tačka se nalazi ispred terminalnog simbola x, ulaz treba popuniti akcijom SHIFT. Izuzetno, ako se radi o nultoj smeni i konfiguraciji  $\langle S' \rangle \rightarrow \langle S \rangle \bullet -$ , akcija je ACCEPT.
2. Ako S sadrži konfiguraciju oblika  $\langle Y \rangle \rightarrow \gamma \bullet$ , x odnosno tačka se nalazi na kraju desne strane, a predikciona komponenta je x, tada ulaz treba popuniti akcijom REDUCE( $\langle Y \rangle \rightarrow \gamma$ ).
3. Vrste koje nisu pokrivene ni 1. ni 2. pravilom treba da sadrže akciju REJECT.

Ukoliko gornja pravila na jednoznačan način određuju akciju za svaki ulaz kontrolne tabele, gramatika pripada klasi LALR(1). U suprotnom, može doći do konflikata tipa potisni-svedi i svedi-svedi, što znači da se gramatika ne može deterministički parsirati LALR(1) parserom. Potisna i kontrolna tabela LALR(1) potisnog automata su prikazane na slici:

	<S>	<A>	<B>	<C>	<D>	a	b	c	a	b	c	-
∇	<S> <sub>0</sub>	<A> <sub>1</sub>	<B> <sub>31</sub>	<C> <sub>4</sub>		a <sub>x</sub>			SHIFT			RED(2)
<S> <sub>0</sub>												ACC
<A> <sub>1</sub>	<S> <sub>1</sub>	<A> <sub>1</sub>	<B> <sub>31</sub>	<C> <sub>4</sub>		a <sub>x</sub>			SHIFT			RED(2)
<B> <sub>31</sub>							b <sub>3</sub>			SHIFT		
<C> <sub>4</sub>								c <sub>4</sub>			SHIFT	
a <sub>x</sub>					<D> <sub>x</sub>					RED(7)	RED(7)	
<S> <sub>1</sub>												RED(1)
b <sub>3</sub>			<B> <sub>32</sub>			a <sub>5</sub>			SHIFT			
c <sub>4</sub>			<B> <sub>4</sub>			a <sub>5</sub>			SHIFT			
<D> <sub>x</sub>										RED(5)	RED(6)	
<B> <sub>33</sub>								c <sub>3</sub>			SHIFT	
a <sub>5</sub>					<D> <sub>5</sub>				RED(7)		RED(7)	RED(7)
<B> <sub>4</sub>									RED(4)			RED(4)
c <sub>3</sub>									RED(3)			RED(3)
<D> <sub>5</sub>									RED(5)		RED(5)	RED(5)

potisna tabela

kontrolna tabela



**Zadatak 5.**

Konstruisati LR(0) i SLR(1) parsere za sledeću gramatiku i prikazati rad automata za sekvencu ((I,I),I)

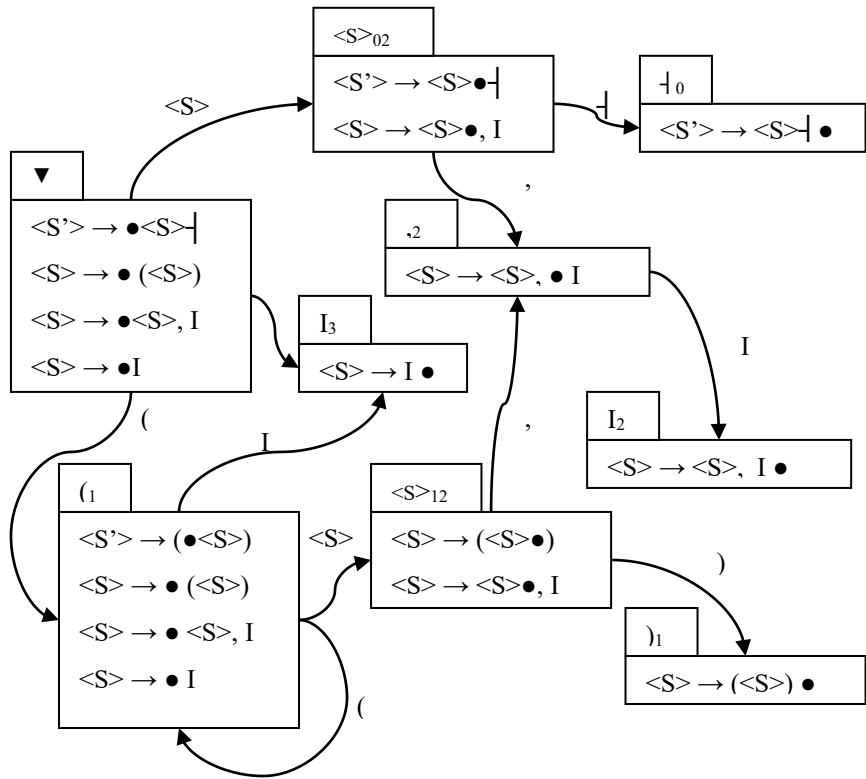
- 1.  $\langle S \rangle \rightarrow (\langle S \rangle)$
- 2.  $\langle S \rangle \rightarrow \langle S \rangle, I$
- 3.  $\langle S \rangle \rightarrow I$

Rešenje:

Proširena gramaika je:

- 0.  $\langle S' \rangle \rightarrow \langle S \rangle \mid$
- 1.  $\langle S \rangle \rightarrow (\langle S \rangle)$
- 2.  $\langle S \rangle \rightarrow \langle S \rangle, I$
- 3.  $\langle S \rangle \rightarrow$

Odgovarajući karakteristični automat je prikazan na slici:



Iz stanja ▼ se preko gramatičkog simbola  $\langle S \rangle$  prelazi u stanje koje ima tačke iza dva neterminala  $\langle S \rangle$  koji odgovaraju neterminalima u smenama 0 i 2. U tom slučaju se stanje ne može obeležiti samo sa  $\langle S \rangle_0$  ili  $\langle S \rangle_2$  kao što se radi u slučaju da je tačka samo iza jednog gramatičkog simbola, nego se obeležava sa  $\langle S \rangle_{02}$  što predstavlja kombinaciju gramatičkih simbola iz smena 0 i 2. Isto se radi za stanje  $\langle S \rangle_{12}$ .

Potisna tabela koja odgovara karakterističnom automatu je:

	(	)	,	I	$\langle S \rangle$	$\mid$
▼	( <sub>1</sub>			I <sub>3</sub>	$\langle S \rangle_{02}$	
$\langle S \rangle_{02}$			, <sub>2</sub>			$\mid_0$
I <sub>3</sub>						
( <sub>1</sub>	( <sub>1</sub>			I <sub>3</sub>	$\langle S \rangle_{12}$	
$\langle S \rangle_{12}$		) <sub>1</sub>	, <sub>2</sub>			
) <sub>1</sub>						
, <sub>2</sub>				I <sub>2</sub>		
$\mid_0$						
I <sub>2</sub>						

**LR(0) parser:**

Kontrolna tabela LR(0) parsera je:

Vrh Steka	bilo koji ulaz
▼	SHIFT
$\langle S \rangle_{02}$	SHIFT
I <sub>3</sub>	REDUCE(3)
( <sub>1</sub>	SHIFT
$\langle S \rangle_{12}$	SHIFT
) <sub>1</sub>	REDUCE(1)
, <sub>2</sub>	SHIFT
$\mid_0$	ACCEPT
I <sub>2</sub>	REDUCE(2)

## SLR(1) parser:

Follow skupovi neterminala i kontrolna tabela je su: Follow ( $\langle S \rangle$ ) =  $\{ \mid \} \text{ ", " }$

	(	)	,	I	$\mid$
▼	SHIFT			SHIFT	
$\langle S \rangle_{02}$			SHIFT		SHIFT
$I_3$		REDUCE(3)	REDUCE(3)		REDUCE(3)
( <sub>1</sub>	SHIFT			SHIFT	
$\langle S \rangle_{12}$		SHIFT	SHIFT		
) <sub>1</sub>		REDUCE(1)	REDUCE(1)		REDUCE(1)
, <sub>2</sub>				SHIFT	
$\mid_0$					ACCEPT
$I_2$		REDUCE(2)	REDUCE(2)		REDUCE(2)

Za ulaz ((I,I)),I

Stek	Ulaz	Akcija
▼	((I,I)),I $\mid$	SHIFT
▼ ( <sub>1</sub>	(I,I)),I $\mid$	SHIFT
▼ ( <sub>1</sub> ( <sub>1</sub>	I,I)),I $\mid$	SHIFT
▼ ( <sub>1</sub> ( <sub>1</sub> <u><math>I_3</math></u>	,I)),I $\mid$	REDUCE(3)
▼ ( <sub>1</sub> ( <sub>1</sub> $\langle S \rangle_{12}$	,I)),I $\mid$	SHIFT
▼ ( <sub>1</sub> ( <sub>1</sub> $\langle S \rangle_{12},_2$	I)),I $\mid$	SHIFT
▼ ( <sub>1</sub> ( <sub>1</sub> <u><math>\langle S \rangle_{12},_2 I_2</math></u>	)),I $\mid$	REDUCE(2)
▼ ( <sub>1</sub> ( <sub>1</sub> $\langle S \rangle_{12}$	)),I $\mid$	SHIFT
▼ ( <sub>1</sub> ( <u><math>I_1 \langle S \rangle_{12} )_1</math></u>	),I $\mid$	REDUCE(1)
▼ ( <sub>1</sub> $\langle S \rangle_{12}$	),I $\mid$	SHIFT
▼ ( <u><math>I_1 \langle S \rangle_{12} )_1</math></u>	,I $\mid$	REDUCE(1)
▼ $\langle S \rangle_{02}$	,I $\mid$	SHIFT
▼ $\langle S \rangle_{02},_2$	I $\mid$	SHIFT
▼ $\langle S \rangle_{02},_2 I_2$	$\mid$	REDUCE(2)
▼ $\langle S \rangle_{02}$	$\mid$	SHIFT
▼ $\langle S \rangle_{02} \mid_0$		ACCEPT

## Zadatak 6.

Konstruisati LR(0) i SLR(1) parsere za sledeću gramatiku i prikazati izgled steka, ulaza i akcije pri procesiranju ulaza „a456b“.

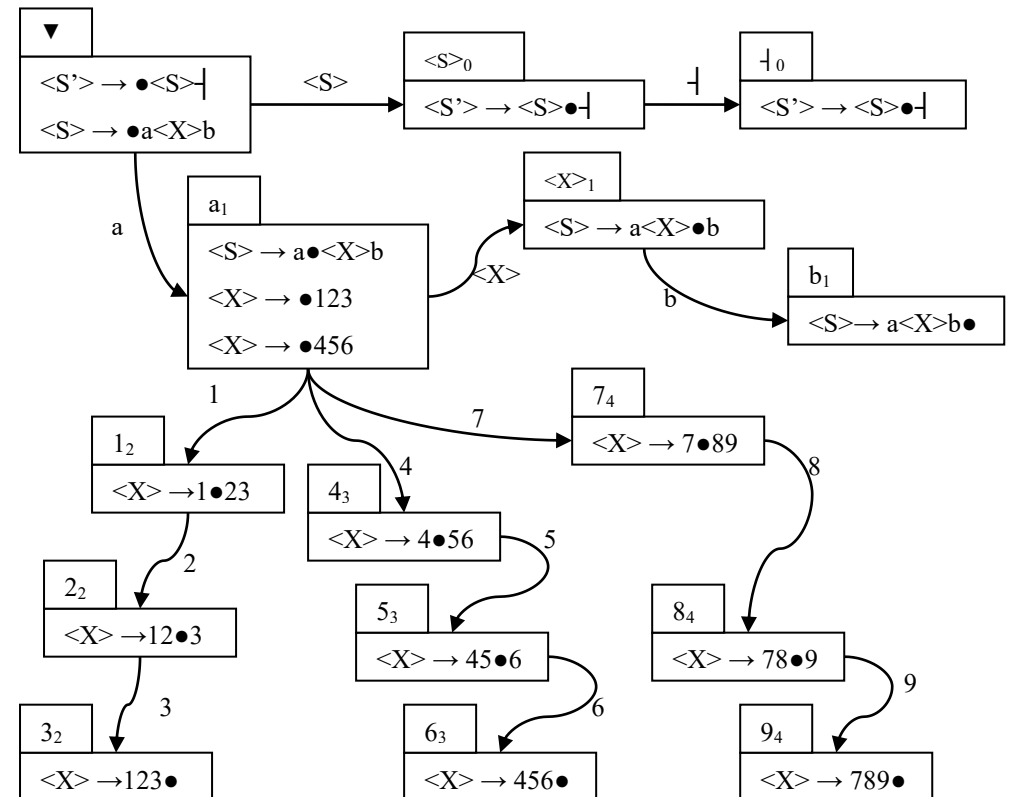
- $\langle S \rangle \rightarrow a \langle X \rangle b$
- $\langle X \rangle \rightarrow 123$
- $\langle X \rangle \rightarrow 456$
- $\langle X \rangle \rightarrow 789$

Rešenje:

Proširena gramatika je prikazana na slici:

- $\langle S' \rangle \rightarrow \langle S \rangle \mid$
- $\langle S \rangle \rightarrow a \langle X \rangle b$
- $\langle X \rangle \rightarrow 123$
- $\langle X \rangle \rightarrow 456$
- $\langle X \rangle \rightarrow 789$

Karakteristični automat koji odgovara gramatici je:



Potisna tabela

LR(0) kontrolna

	<S>	a	<X>	b	1	2	3	4	5	6	7	8	9		Bilo koji ulaz
▼	<S> <sub>0</sub>	a <sub>1</sub>													SHIFT
<S> <sub>0</sub>														<sub>0</sub>	SHIFT
<sub>0</sub>															ACCEPT
a <sub>1</sub>			<X> <sub>1</sub>		1 <sub>2</sub>			4 <sub>3</sub>			7 <sub>4</sub>				SHIFT
<X> <sub>1</sub>				b <sub>1</sub>											SHIFT
b <sub>1</sub>															RED(1)
1 <sub>2</sub>						2 <sub>2</sub>									SHIFT
2 <sub>2</sub>							3 <sub>2</sub>								SHIFT
3 <sub>2</sub>															RED(2)
4 <sub>3</sub>									5 <sub>3</sub>						SHIFT
5 <sub>3</sub>										6 <sub>3</sub>					SHIFT
6 <sub>3</sub>															RED(3)
7 <sub>4</sub>												8 <sub>4</sub>			SHIFT
8 <sub>4</sub>													9 <sub>4</sub>		SHIFT
9 <sub>4</sub>															RED(4)

SLR(1) parser kontrolna tabela:

Follow (&lt;S&gt;) = |    Follow (&lt;X&gt;) = b

	a	b	1	2	3	4	5	6	7	8	9	
▼	S											
<S> <sub>0</sub>												S
<sub>0</sub>												AC
a <sub>1</sub>			S			S			S			
<X> <sub>1</sub>		S										
b <sub>1</sub>												R1
1 <sub>2</sub>				S								
2 <sub>2</sub>					S							
3 <sub>2</sub>		R2										
4 <sub>3</sub>							S					
5 <sub>3</sub>								S				
6 <sub>3</sub>		R3										
7 <sub>4</sub>										S		
8 <sub>4</sub>											S	
9 <sub>4</sub>		R4										

Za ulaz „a456b“ stek, ulaz i akcije koje se vrše su:

Stek	Ulaz	Akcija
▼	a456b	SHIFT
▼ a <sub>1</sub>	456b	SHIFT
▼ a <sub>1</sub> 4 <sub>3</sub>	56b	SHIFT
▼ a <sub>1</sub> 4 <sub>3</sub> 5 <sub>3</sub>	6b	SHIFT
▼ a <sub>1</sub> <u>4<sub>3</sub> 5<sub>3</sub> 6<sub>3</sub></u>	b	REDUCE(3)
▼ a <sub>1</sub> <X> <sub>1</sub>	b	SHIFT
▼ <u>a<sub>1</sub> &lt;X&gt;<sub>1</sub> b<sub>1</sub></u>		REDUCE(1)
▼ <S> <sub>0</sub>		SHIFT
▼ <S> <sub>0</sub>   <sub>0</sub>		ACCEPT

**Zadatak 7.**

Za gramatiku na slici konstruisati LR(0), SLR(1) i LALR(1) parsere.

1.  $\langle S \rangle \rightarrow b\langle A \rangle$
2.  $\langle S \rangle \rightarrow \langle X \rangle a$
3.  $\langle X \rangle \rightarrow \varepsilon$
4.  $\langle A \rangle \rightarrow \langle X \rangle$

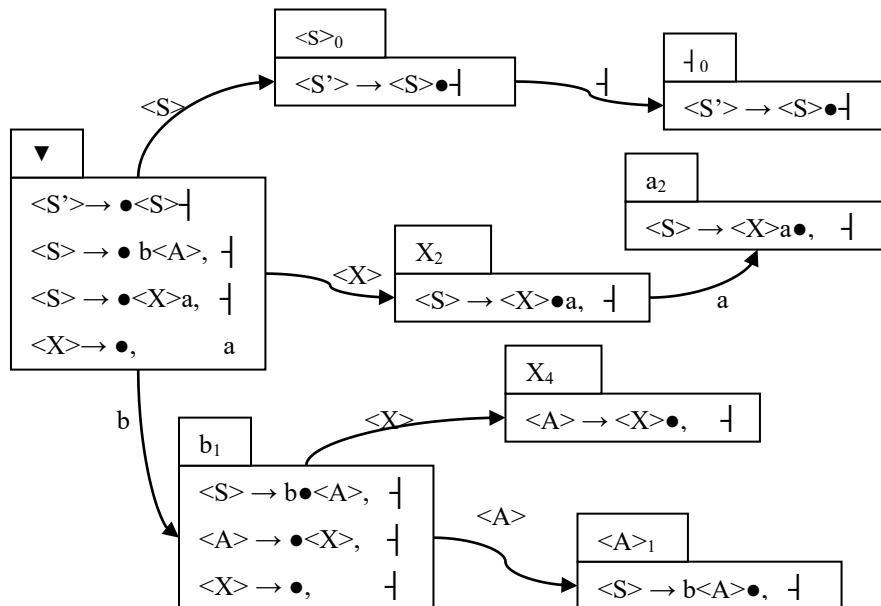
Prikazati rad SLR(1) i LALR(1) parsera za ulaz „aa“.

*Rešenje:*

Proširena gramatika je prikazana na slici:

0.  $\langle S' \rangle \rightarrow \langle S \rangle$
1.  $\langle S \rangle \rightarrow b \langle A \rangle$
2.  $\langle S \rangle \rightarrow \langle X \rangle a$
3.  $\langle X \rangle \rightarrow \epsilon$
4.  $\langle A \rangle \rightarrow \langle X \rangle$

Graf karakterističnog automata je identičan za LR(0), SLR(1) i LALR(1) automate. Za LR(0) i SLR(1) automate se ne koriste predikcioni simboli.



Potisna tabela je:

	$\langle S \rangle$	$\vdash$	b	$\langle X \rangle$	a	$\langle A \rangle$	Bilo koji ulaz
$\nabla$	$\langle S \rangle_0$		$b_1$	$\langle X \rangle_2$			REDUCE(3)/SHIFT
$\langle S \rangle_0$		$\vdash_0$					SHIFT
$\vdash_0$							ACCEPT
$b_1$				$\langle X \rangle_4$		$\langle A \rangle_1$	REDUCE(3)/SHIFT
$\langle A \rangle_1$							REDUCE(1)
$a_2$							REDUCE(2)
$\langle X \rangle_2$					$a_2$		SHIFT
$\langle X \rangle_4$							REDUCE(4)

Kao što se vidi SHIFT/REDUCE se pojavljuju u ćelijama tabele. LR(0) parser ne može da se konstruiše u slučaju da se u gramatici nalaze prazne smene.

## SLR(1) parser

Follow skupovi neterminala su:

$$\text{Follow}(\langle S \rangle) = \mid$$
$$\text{Follow}(\langle A \rangle) = \{$$
$$\text{Follow}(\langle X \rangle) = a \mid$$

Kontrolna tabela SLR(1) parsera je:

	$\downarrow$	b	a
$\blacktriangledown$	REDUCE(3)	SHIFT	REDUCE(3)
$\langle S \rangle_0$	SHIFT		
$\downarrow_0$	ACCEPT		
$b_1$	REDUCE(3)		REDUCE(3)
$\langle A \rangle_1$	REDUCE(1)		
$a_2$	REDUCE(2)		
$\langle X \rangle_2$			SHIFT
$\langle X \rangle_4$	REDUCE(4)		

Prilikom obrade ulaza „aa“ vrše se sledeće akcije:

Stek	Ulaz	Akcija
▼	aa	REDUCE(3)
▼ <X> <sub>2</sub>	aa	SHIFT
▼ <X> <sub>2</sub> a <sub>2</sub>	a	REJECT

LALR(1) parser – kontrolna tabela:

		b	a
▼		SHIFT	REDUCE(3)
<S> <sub>0</sub>	SHIFT		
<sub>0</sub>	ACCEPT		
b <sub>1</sub>	REDUCE(3)		
<A> <sub>1</sub>	REDUCE(1)		
a <sub>2</sub>	REDUCE(2)		
<X> <sub>2</sub>			SHIFT
<X> <sub>4</sub>	REDUCE(4)		

Kao što se vidi, za razliku od SLR(1) automata, u ćelijama (▼, |), (<A>, |), (<A>, b) i (a<sub>2</sub>, a) se ne nalaze REDUCE akcije nego REJECT akcije. Zbog restriktivnijih pravila LALR(1) automati su „otporniji“ na SHIFT/REDUCE konflikte i generalno brže otkrivaju greške od SLR(1) automata. Pri tome su karakteristični automati istih veličina.

Prilikom obrade ulaza „aa“ vrše se sledeće akcije:

Stek	Ulaz	Akcija
▼	aa	REDUCE(3)
▼ <X> <sub>2</sub>	aa	SHIFT
▼ <X> <sub>2</sub> a <sub>2</sub>	a	REJECT

LALR(1) i SLR(1) parseri imaju identične karakteristične automate, ali LALR(1) automat brže nalazi greške. Za praznu sekvencu, LALR(1) automat bi u prvom koraku našao grešku, dok bi SLR(1) automat grešku našao tek u drugom koraku.

### Zadatak 8.

Za gramatiku na slici konstruisati LALR(1) parser i prikazati akcije parsera(stek, ulaz i akcije) za ulaz aIa.

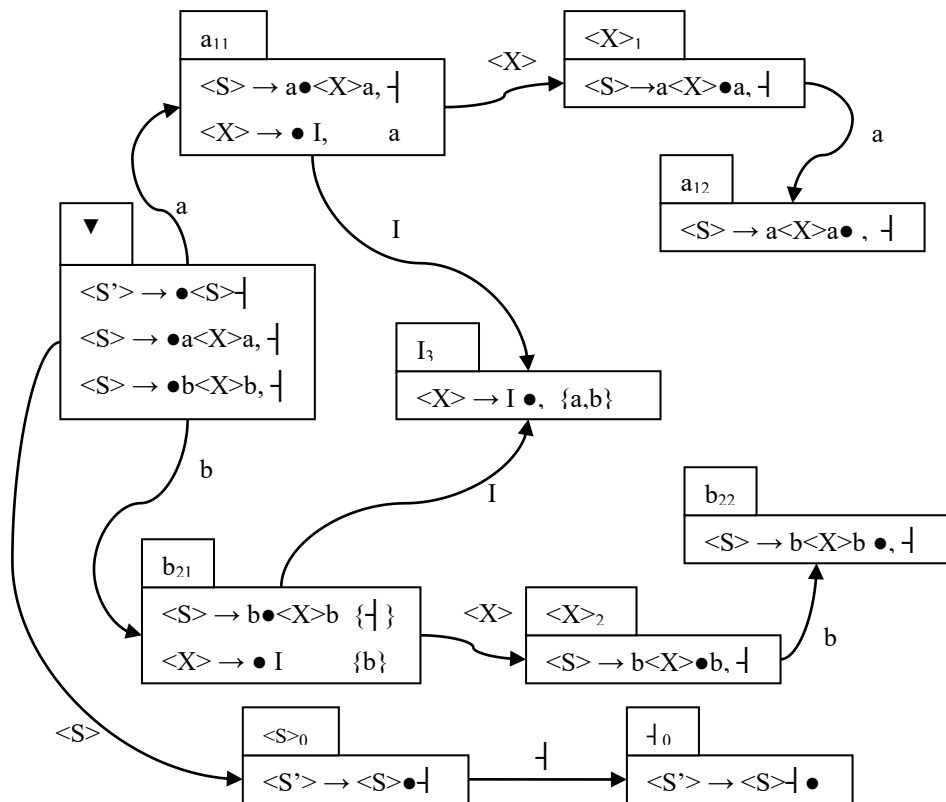
1.  $\langle S \rangle \rightarrow a\langle X \rangle a$
2.  $\langle S \rangle \rightarrow b\langle X \rangle b$
3.  $\langle X \rangle \rightarrow I$

Rešenje:

Proširena gramatika:

0.  $\langle S' \rangle \rightarrow \langle S \rangle \mid$
1.  $\langle S \rangle \rightarrow a\langle X \rangle a$
2.  $\langle S \rangle \rightarrow b\langle X \rangle b$
3.  $\langle X \rangle \rightarrow I$

Karakteristični LALR(1) automat je prikazan na slici:



Potisna tabela je :

LALR(1) kontrolna tabela

	$\langle S \rangle$	$\langle X \rangle$	a	b	I	$\mid$	a	b	I	$\mid$
$\nabla$	$\langle S \rangle_0$		$a_{11}$	$b_{21}$			SH	SH		
$\langle S \rangle_0$						$\mid_0$				SH
$\mid_0$										ACC
$a_{11}$		$\langle X \rangle_1$			$I_3$				SH	
$\langle X \rangle_1$			$a_{12}$				SH			
$a_{12}$										R(1)
$b_{21}$		$\langle X \rangle_2$			$I_3$				SH	
$\langle X \rangle_2$				$b_{22}$				SH		
$b_{22}$										R(2)
$I_3$							R(3)	R(3)		

potisnog LALR(1) automata je:

Za ulaz „aIa“ stek, ulaz i akcije koje se vrše su:

Stek	Ulaz	Akcija
$\nabla$	aIa $\mid$	SHIFT
$\nabla a_{11}$	Ia $\mid$	SHIFT
$\nabla a_{11} I_3$	a $\mid$	REDUCE(3)
$\nabla a_{11} \langle X \rangle_1$	a $\mid$	SHIFT
$\nabla a_{11} \langle X \rangle_1 a_{12}$	$\mid$	REDUCE(1)
$\nabla \langle S \rangle_0$	$\mid$	SHIFT
$\nabla \langle S \rangle_0 \mid_0$		ACCEPT

Kao što se vidi LR(1) i LALR(1) parseri imaju slične akcije za isti ulaz. Jedina razlika je u činjenici da LALR(1) parser ima manje stanja u karakterističnom automatu.

### Zadatak 9.

Za gramatiku na slici prikazati kako se određuju predikcioni simboli LALR(1) automata u stanju ▼.

1.  $\langle S \rangle \rightarrow \langle A \rangle \langle X \rangle$
2.  $\langle A \rangle \rightarrow \langle B \rangle \langle Y \rangle$
3.  $\langle B \rangle \rightarrow \langle C \rangle \langle Z \rangle$
4.  $\langle C \rangle \rightarrow c$
5.  $\langle X \rangle \rightarrow x \mid \varepsilon$
6.  $\langle Y \rangle \rightarrow y \mid \varepsilon$
7.  $\langle Z \rangle \rightarrow z \mid \varepsilon$

Rešenje:

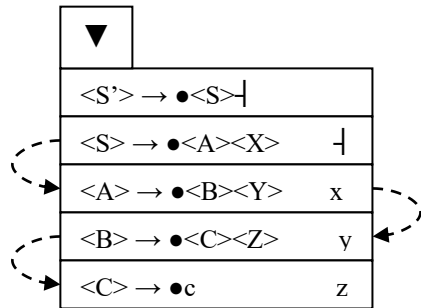
Proširena gramatika i startno stanje karakterističnog LR(0) automata su prikazani na slici:

0.  $\langle S' \rangle \rightarrow \langle S \rangle \mid$
1.  $\langle S \rangle \rightarrow \langle A \rangle \langle X \rangle$
2.  $\langle A \rangle \rightarrow \langle B \rangle \langle Y \rangle$
3.  $\langle B \rangle \rightarrow \langle C \rangle \langle Z \rangle$
4.  $\langle C \rangle \rightarrow c$
5.  $\langle X \rangle \rightarrow x \mid \varepsilon$
6.  $\langle Y \rangle \rightarrow y \mid \varepsilon$
7.  $\langle Z \rangle \rightarrow z \mid \varepsilon$

SLA skupovi konfiguracija su:

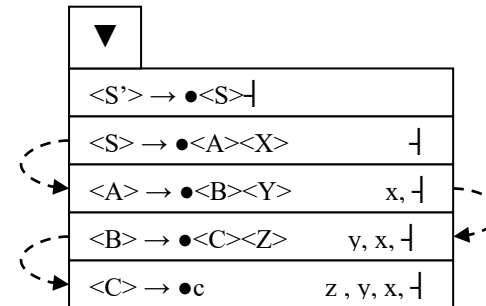
- SLA( $\langle S \rangle \rightarrow \bullet \langle A \rangle \langle X \rangle$ ) =  $\mid$ , zato što je  $\mid$  iza  $\langle S \rangle$  u  $\langle S' \rangle \rightarrow \bullet \langle S \rangle \mid$   
 SLA( $\langle A \rangle \rightarrow \bullet \langle B \rangle \langle Y \rangle$ ) =  $x$ , zato što je  $\langle X \rangle$  iza  $\langle A \rangle$  u  $\langle S \rangle \rightarrow \bullet \langle A \rangle \langle X \rangle$ ,  
 i  $\text{First}(\langle X \rangle) = x$   
 SLA( $\langle B \rangle \rightarrow \bullet \langle C \rangle \langle Z \rangle$ ) =  $y$ , zato što je  $\langle Y \rangle$  iza  $\langle B \rangle$  u  $\langle A \rangle \rightarrow \bullet \langle B \rangle \langle Y \rangle$   
 i  $\text{First}(\langle Y \rangle) = y$   
 SLA( $\langle C \rangle \rightarrow \bullet c$ ) =  $z$ , zato što je  $\langle Z \rangle$  iza  $\langle C \rangle$  u  $\langle B \rangle \rightarrow \bullet \langle C \rangle \langle Z \rangle$   
 i  $\text{First}(\langle Z \rangle) = z$

Na slici su prikazane konfiguracije sa SLA skupovima i vezama prosleđivanja. Veze prosleđivanja su obeležene isprekidanim linijama.



1. Veza od prve ( $\langle S \rangle \rightarrow \bullet \langle A \rangle \langle X \rangle$ ) do druge konfiguracije ( $\langle A \rangle \rightarrow \bullet \langle B \rangle \langle Y \rangle$ ) postoji zato što je  $\langle X \rangle$  poništivo u prvoj konfiguraciji
2. Veza od druge ( $\langle A \rangle \rightarrow \bullet \langle B \rangle \langle Y \rangle$ ) do treće konfiguracije ( $\langle B \rangle \rightarrow \bullet \langle C \rangle \langle Z \rangle$ ) postoji zato što je  $\langle Y \rangle$  poništivo u drugoj konfiguraciji
3. Veza od treće ( $\langle B \rangle \rightarrow \bullet \langle C \rangle \langle Z \rangle$ ) do četvrte konfiguracije ( $\langle C \rangle \rightarrow \bullet c$ ) postoji zato što je  $\langle Z \rangle$  poništivo u prvoj konfiguraciji

Kada se odrede SLA skupovi, njihove vrednosti propagiraju "u smeru strelica" veza prosleđivanja. Na ovaj način u skup predikcionih simbola druge konfiguracije je dodat terminal  $\mid$ , u skup predikcionih simbola treće konfiguracije su dodati  $x$  i  $\mid$  iz druge konfiguracije, i u skup predikcionih simbola četvrte konfiguracije su dodati terminali  $y$ ,  $x$  i  $\mid$  iz treće konfiguracije. Na ovaj nači su određeni predikcioni skupovi konfiguracija u stanju ▼. Konačni predikcioni skupovi konfiguracija su prikazani na slici:



Za gornju gramatiku u stanju  $\nabla$  jedina veza prosleđivanja je iz druge ka trećoj konfiguraciji, pošto je jedino neterminal  $\langle E \rangle$  poništiv.



# Oporavak od grešaka

## Zadatak 1.

Prikazati izgled steka u slučaju da se na ulaz CUP generisanog parsera koji je opisan datom gramatikom dovede ulaz  $a=b+d+c$ .

1.  $\langle S \rangle \rightarrow \langle S \rangle \langle I \rangle$

2.  $\langle S \rangle \rightarrow \langle I \rangle$
3.  $\langle I \rangle \rightarrow a=b+c;$

4.  $\langle I \rangle \rightarrow a=error;$

### Rešenje

U specifikacijama parsera od dna ka vrhu često se postavljaju smene u kojima se definiše kako se parser ponaša u slučaju da naiđe na neočekivani token. Ako parser naiđe na grešku u cilju oporavka primenjuje se sledeći postupak:

1. Sa steka se skidaju elementi dok se ne naiđe na element posle kog se može staviti error symbol. Ako se ne nađe takav element parser ne može da se oporavi od greške.
2. Na stek se postavlja simbol error
3. Na ulazu se preskaču simboli dok se ne naiđe na simbol koji može da se postavi na error simbol
4. Kada se kompletira ručka vrši se redukcija vrha steka i parsiranje se nastavlja.

Izgled steka i ulaza je prikazano na slici:

Stanje steka :	Ulazni simbol :	Akcija
1. $\nabla$	$a=b+d+c; \mid$	SHIFT
2. $\nabla a$	$=b+d+c; \mid$	SHIFT
3. $\nabla a=$	$b+d+c; \mid$	SHIFT
4. $\nabla a=b$	$+d+c; \mid$	SHIFT
5. $\nabla a=b+$	$d+c; \mid$	POP
6. $\nabla a=b$	$d+c; \mid$	POP
7. $\nabla a=$	$d+c; \mid$	SHIFT(error)
8. $\nabla a=error$	$d+c; \mid$	ADVANCE
9. $\nabla a=error$	$+c; \mid$	ADVANCE
10. $\nabla a=error$	$c; \mid$	ADVANCE
11. $\nabla a=error$	$; \mid$	SHIFT
12. $\nabla \underline{a=error;}$	$\mid$	REDUCE(4)
13. $\nabla \underline{\langle I \rangle}$	$\mid$	REDUCE(2)
14. $\nabla \langle S \rangle$	$\mid$	ACCEPT

## Zadatak 2.

Prikazati izgled steka u slučaju da se na ulaz parsera koji parsira od dna ka vrhu koji je opisan datom gramatikom

1.  $\langle S \rangle \rightarrow \langle S \rangle \langle I \rangle$
2.  $\langle S \rangle \rightarrow \langle I \rangle$
3.  $\langle I \rangle \rightarrow v = v + c;$
4.  $\langle I \rangle \rightarrow v = c;$
5.  $\langle I \rangle \rightarrow \text{if } \langle E \rangle \text{ then } \langle S \rangle \text{ endif};$
6.  $\langle E \rangle \rightarrow v = c$

dovede ulaz:

```
a=1;
if a==0 then
    b=1%
    if b==3 then x=a+1;
    endif;
    x=2;
endif;
```

a) Pretpostavka je da se za oporavak od grešaka se koristi jednostavan panic mode algoritam sa sigurnim simbolima ; i endif.

b) Pretpostavka je da se za oporavak od grešaka se koristi poboljšani panic mode algoritam sa simbolom zaglavlja if i sigurnim simbolima ; i endif.

### Rešenje

Kada parser na ulazu naiđe na grešku pokušava da se oporavi od nje i nastavlja parsiranje (oporavak od greške). U slučaju da oporavak nije ispravno odabran parser nailazi na nove ispravan kod prijavljuje kao pogrešan. U gornjem primeru parser nailazi na grešku posle konstrukcije  $b=a$  pošto iza izraza nema tačke-zarez već nelegalni znak %.

a) Jednostavan panic mode algoritam glasi:

- (1) Kada parser detektuje grešku, sa ulaza se redom uklanjaju simboli dok se ne naiđe na neki od sigurnih simbola.
- (2) Tada se sa steka parsera skidaju simboli sve dok se ne naiđe na neki koji može da potisne sigurni simbol na ulazu na stek.
- (3) Parsiranje se posle toga nastavlja na uobičajen način.

Stanje steka :	Preostali ulaz:	Akcija
1. $\nabla$	a=1;...	SH,SH,SH,SH
2. $\nabla \underline{v=c;}$	if a==0...	REDUCE(4)
3. $\nabla \underline{\langle I \rangle}$	if a==0...	REDUCE(2)
4. $\nabla \langle S \rangle$	if a==0...	SH,SH,SH,SH,SH
5. $\nabla \langle S \rangle \text{ if } \underline{v=c}$	then b=1%...	REDUCE(6)
6. $\nabla \langle S \rangle \text{ if } \langle E \rangle$	then b=1%...	SH,SH,SH,SH
7. $\nabla \langle S \rangle \text{ if } \langle E \rangle \text{ then } v=c$	%if b==3 then x=a+1;...	PNC1: Advance 11
8. $\nabla \langle S \rangle \text{ if } \langle E \rangle \text{ then } v=c$	; endif;x=2;...	PNC2: SH
9. $\nabla \langle S \rangle \text{ if } \langle E \rangle \text{ then } \underline{v=c;}$	endif; x=2;...	REDUCE(4)
10. $\nabla \langle S \rangle \text{ if } \langle E \rangle \text{ then } \langle I \rangle$	endif; x=2;...	REDUCE(2)
11. $\nabla \langle S \rangle \text{ if } \langle E \rangle \text{ then } \langle S \rangle$	endif; x=2;...	SH,SH
12. $\nabla \langle S \rangle \underline{\text{if } \langle E \rangle \text{ then } \langle S \rangle \text{ endif};}$	x=2;...	REDUCE(5)
13. $\nabla \underline{\langle S \rangle \langle I \rangle}$	x=2;...	REDUCE(1)
14. $\nabla \langle S \rangle$	x=2;...	SH,SH,SH,SH
15. $\nabla \langle S \rangle \underline{v=c;}$	endif;...	REDUCE(4)
16. $\nabla \underline{\langle S \rangle \langle I \rangle}$	endif;...	REDUCE(1)
17. $\nabla \langle S \rangle$	endif;...	PNC1:-
18. $\nabla \langle S \rangle$	endif...	PNC2: POP
19. $\nabla$	endif...	REJECT

Parser nije uspeo da se oporavi od greške. Osnovni problem je što je preskočio početak if-a u koraku 7 tako da je pogrešno upario spoljni if sa unutrašnjim endif.

b) Poboljšani algoritam oporavka glasi:

- (1) Kada parser detektuje grešku, sa ulaza se redom uklanjaju simboli dok se ne naiđe na neki od sigurnih simbola ili simbola zaglavlja.
- (2a) Ako je naišao siguran simbol, tada se sa steka parsera skidaju simboli sve dok se ne naiđe na neki koji može da potisne sigurni simbol na ulazu na stek.
- (2b) Ako je naišao simbol zaglavlja, tada se on potiskuje na stek i posebno markira.
- (3) Parsiranje se posle toga nastavlja na uobičajen način. Kada se redukcijom neke smene ukloni markirani simbol sa steka, prelazi se ponovo na korak (1).

Stanje steka :	Preostali ulaz:	Akcija
1. $\nabla$	a=1;...	SH,SH,SH,SH
2. $\nabla$ <u>v=c;</u>	if a==0...	REDUCE(4)
3. $\nabla$ <u>&lt;I&gt;</u>	if a==0...	REDUCE(2)
4. $\nabla$ <S>	ifa==0...	SH,SH,SH,SH,SH
5. $\nabla$ <S> if <u>v==c</u>	then b=1%...	REDUCE(6)
6. $\nabla$ <S> if <E>	then b=1%...	SH,SH,SH,SH
7. $\nabla$ <S> if <E> then v=c	%...	PNC1: Advance 1
8. $\nabla$ <S> if <E> then v=c	if...	PNC2b: SH
9. $\nabla$ <S> if <E> then v=c <b>if</b>	b==3...	PNC3: SH,SH,SH
10. $\nabla$ <S> if <E> then v=c <b>if</b> <u>v==c</u>	then x=a+1;...	REDUCE(6)
11. $\nabla$ <S> if <E> then v=c <b>if</b> <E>	then x=a+1;...	SH 7x
12. $\nabla$ <S> if <E> then v=c <b>if</b> <E> then <u>v=v+c;</u>	endif;...	REDUCE(3)
13. $\nabla$ <S> if <E> then v=c <b>if</b> <E> then <I>	endif;...	SH,SH
14. $\nabla$ <S> if <E> then v=c <b>if</b> <E> <b>then</b> <I> <b>endif</b> ;	...	REDUCE(5)
15. $\nabla$ <S> if <E> then v=c <I>	x=2;...	PNC1: Advance 3
16. $\nabla$ <S> if <E> then v=c <I>	;...	PNC2a: POP
17. $\nabla$ <S> if <E> then v=c	;...	SH
18. $\nabla$ <S> if <E> then <u>v=c</u> ;	endif;	REDUCE(4)
19. $\nabla$ <S> if <E> then <u>&lt;I&gt;</u>	endif;	REDUCE(2)
20. $\nabla$ <S> if <E> then <S>	endif;	SH,SH
21. $\nabla$ <S> <b>if</b> <E> <b>then</b> <S> <b>endif</b> ;		REDUCE(5)
22. $\nabla$ <S> <I>		REDUCE(1)
23. $\nabla$ <S>		SH
24. $\nabla$ <S>—		ACCEPT

Poboljšani algoritam uspeo je da se oporavi od greške u ovom slučaju.